

SMS directe

Aspects generals de l'API

Índex

1. Introducció	3
2. Aspectes generals de l'API	4
2.1. Peticions en format XML	4
2.2. Peticions en format JSON	5
2.3. Peticions amb variables	5
2.4. Formats de resposta	6
2.5. Codis d'estat	8
2.6. Exemples de petició (PHP i JAVA)	9

1. Introducció

2. Aspectes generals de l'API

La forma més senzilla d'autenticació és usant el nom d'usuari en el paràmetre `user` i la contrasenya en el paràmetre `password`.

En els exemples d'aquesta documentació es fa servir el símbol `{:name}` per al nom d'usuari i `{:pass}` per a la contrasenya.

Totes les operacions requereixen els paràmetres següents:

- **user** Nom d'usuari del compte a la plataforma SMS directe.
- **password** La contrasenya de l'usuari.
- **request** Tipus de petició.
- **request_id** *Opcional*. Identificador de la petició. Encara que sigui opcional en recomanem l'ús.

Depenent de quina API s'estigui utilitzant, els paràmetres d'entrada poden estar encapsulats en un document XML o JSON o bé amb variables. Aquest document (XML/JSON) ha de seguir el format definit en el dtd de cada API.

2.1. Peticions en format XML

```
GET https://apismsdirecte.andorratelecom.ad/action/?xml={:action_xml}  
HTTP/1.1  
Accept: application/xml
```

```
POST https://apismsdirecte.andorratelecom.ad/action/ HTTP/1.1  
Content-type: application/xml; charset=ISO-8859-1  
Accept: application/xml
```

```
xml={:action_xml}
```

2.2. Peticions en format JSON

```
GET https://apismsdirecte.andorratelecom.ad/action/?json={:action_
json} HTTP/1.1
Accept: application/json
```

```
POST https://apismsdirecte.andorratelecom.ad/action/ HTTP/1.1
Content-type: application/json; charset=utf-8
Accept: application/json

json={:action_json}
```

2.3. Peticions amb variables

Una altra alternativa és enviar els paràmetres com a tuples key:value. La resposta es pot configurar tant en format XML com JSON.

```
GET https://apismsdirecte.andorratelecom.ad
action/?user={:name}&password={:pass}&... HTTP/1.1
```

```
POST https://apismsdirecte.andorratelecom.ad/action/ HTTP/1.1
Content-type: application/x-www-form-urlencoded
Accept: application/xml

user={:name}&password={:pass}&...
```

2.4. Formats de resposta

Totes les operacions reben com a resposta un document XML o JSON, el node del qual és l'element `response` o `result` depenent de l'API en ús. Aquest es compon de diversos elements comuns i sempre presents en totes les operacions, seguits d'altres elements específics per a l'operació que s'ha invocat.

Els elements comuns de l'element `response` són els següents:

- **code** És el codi d'estat informatiu de l'èxit o error de l'operació.
- **status** És la descripció del codi d'estat.
- **request** Conté un nom identificatiu de l'operació que s'ha invocat.

Els elements comuns de l'element `result` són els següents:

- **action** Tipus de consulta realitzada.
- **status** Codi d'estat de la consulta.
- **msg** Descripció del codi d'estat.

Les respostes compleixen amb l'especificació de la DTD. Aquestes canvien depenent de quina API estigui en ús.

A continuació, es mostra un exemple d'una consulta que no ha tingut èxit amb la corresponent resposta, d'una API en què el node arrel és tant `response` com `result`. Com que és una consulta fallida, la resposta només conté els elements comuns.

Node arrel de resposta `result`

Consulta:

```
<? xml version="1.0" encoding="ISO-8859-1" ?>
<mo_download_pdf>
  <user>meusuari</user>
  <password>mevapasswd</password>
  <src>+376600000</src>
  <dst>+376</dst>
  <mo>99999999999</mo>
</mo_download_pdf>
```

Resposta:

```
<? xml version="1.0" encoding="ISO-8859-1" ?>
<result>
  <action>download_pdf</action>
  <status>-3</status>
  <msg>Invalid query</msg>
</result>
```

Node arrel de resposta *response*

Consulta:

```
<? xml version="1.0" encoding="ISO-8859-1" ?>
<info>
  <user>meusuari</user>
  <password>mevapasswd</password>
</info>
```

Resposta:

```
<? xml version="1.0" encoding="ISO-8859-1" ?>
<response>
  <request>info</request>
  <code>1600</code>
  <status>Invalid user</status>
</response>
```

2.5. Codis d'estat

Estat	Descripció	Estat	Descripció
200	Success	1606	Invalid number
1500	Unknown error	1607	Number not available
1501	Timeout error	1608	Prefix not available
1502	Connection error	1610	Invalid password
1503	Service unavailable	1611	Invalid datetime
1504	Temporary error. Try again	1700	Invalid provider
1506	Not implemented yet	1701	Invalid customer phone number
1507	Try again later	1801	Invalid event
1400	Method not allowed	1802	Invalid action
1401	Invalid request	1803	Duplicated action
1402	Invalid content	1901	Invalid source
1404	Invalid XML	1902	Invalid destinationary
1405	Invalid JSON	1903	Invalid message
1600	Invalid user	1904	MT ID not vaid
1601	No credit	3001	Invalid subject
1603	Invalid id	4001	Error input stream
1604	ID Not found	4004	Error file type
1605	Invalid mail	4006	File does not exist

2.6. Exemples de petició (PHP i JAVA)

PHP

```
$url = 'https://apismsdirecte.andorratelecom.ad/sms/';
$xml = '<sms><user>'.LOGIN.'</user><password>'.PASSW.'</password><dst><num>'.NUM.
      '</num></dst><txt>'.TXT.'</txt></sms>';
$xml='xml='.$xml;

$opciones = array('http' => array('method' => 'POST',
                                'header' => 'Content-type: application/x-www-form-
urlencoded',
                                'content' => $xml));
$contexto = stream_context_create($opciones);
$resultado = file_get_contents($url, false, $contexto);
```

Utilitzant curl

```
$url = 'https://apismsdirecte.andorratelecom.ad/sms/';
$xml = '<sms><user>'.LOGIN.'</user><password>'.PASSW.'</password><dst><num>'.NUM.
      '</num></dst><txt>'.TXT.'</txt></sms>';
$xml = 'xml='.$xml;

$sh = curl_init($url);
curl_setopt($sh, CURLOPT_POST, 1);
curl_setopt($sh, CURLOPT_POSTFIELDS, $xml);
curl_setopt($sh, CURLOPT_HEADER, 0);
curl_setopt($sh, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($sh, CURLOPT_RETURNTRANSFER, 1);

// Aprovecho la variable $xml!
$xml = curl_exec($sh);
curl_close($sh);
```

Java

```
import java.io.File;
import java.io.IOException;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.mime.HttpMultipartMode;
import org.apache.http.entity.mime.MultipartEntityBuilder;
import org.apache.http.entity.mime.content.FileBody;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.util.EntityUtils;

public class FinDEVr2k16_request_send {
    public static void main(String[] args) throws ClientProtocolException, IOException
    {

        String user = "user@andorratelecom" ;
        String password = "password";
        String request = "send";
        String connectaclic_id = "" ;
        String recipients = "address@myclient.com";
        String to_registered = "contracts@mycompany.com";
        String to_status = "status@mycompany.com";
        String send_stamp = "1" ;

        CloseableHttpClient httpClient = HttpClientBuilder.create().build();
        HttpPost post = new HttpPost("https://apisms.andorratelecom.ad/connectaclic/v1/basic");
        MultipartEntityBuilder builder = MultipartEntityBuilder.create();
        builder.setMode(HttpMultipartMode.BROWSER_COMPATIBLE);

        builder.addTextBody("user", user);
        builder.addTextBody("password", password);
        builder.addTextBody("request", request);

        builder.addTextBody("connectaclic_id", connectaclic_id);
        builder.addTextBody("recipients", recipients);
        builder.addTextBody("to_registered", to_registered);

        builder.addTextBody("to_status", to_status);
        builder.addTextBody("send_stamp", send_stamp);

        File f = new File("/home/local/SMSdirecte/hero1.pdf");

        builder.addPart("file",new FileBody(f, "application/pdf"));

        post.setEntity(builder.build());
        HttpResponse response = httpClient.execute(post);
        HttpEntity entity = response.getEntity();

        System.out.println( entity) ;
        System.out.println( EntityUtils.toString(entity));

        EntityUtils.consume(entity);
        httpClient.close();
    }
}
```

